Statistics/Data analysis

```
      name:  <unnamed>
       log:  O:\Fall 2021\U of A\MET HW\Final Project\Adopo_Jeffries_Final_log.smcl
  log type:  smcl
 opened on:  17 Dec 2021, 09:28:22

1 .
2 . /* READ ME
  >
  > University of Arkansas, Fayetteville, Arkansas
  > Walton Graduate School of Business
  > ECON 5783, Applied Microeconometrics
  >
  > For: Course Project
  > Written by: Daryl Adopo & Robby Jeffries
  > Date: 17 December 2021
  >
  > */
3 .
4 .
5 .
6 .
7 .
8 . **#
9 . // ---------------------------------------------------- (0) Installing packages and
  > graph schemes
10.
11. local install = 1 // Change this to 1 to download packages/schemes for this project

12.
13. if `install' == 1 {
14.         ssc install rdrobust, replace // Includes many useful commands for RDD
  checking rdrobust consistency and verifying not already installed...
  installing into C:\Users\rejeffri\ado\plus\...
  installation complete.
15.         ssc install estout, replace // For storing and outputting results
  checking estout consistency and verifying not already installed...
  installing into C:\Users\rejeffri\ado\plus\...
  installation complete.
16.         net install rddensity, from("https://raw.githubusercontent.com/rdpackages/rd
  > density/master/stata") replace
  checking rddensity consistency and verifying not already installed...
  installing into C:\Users\rejeffri\ado\plus\...
  installation complete.
17.         net install lpdensity, from("https://raw.githubusercontent.com/nppackages/lp
  > density/master/stata") replace
  checking lpdensity consistency and verifying not already installed...
  installing into C:\Users\rejeffri\ado\plus\...
  installation complete.
18.         net install rdpower, from("https://raw.githubusercontent.com/rdpackages/rdpo
  > wer/master/stata") replace
  checking rdpower consistency and verifying not already installed...
  installing into C:\Users\rejeffri\ado\plus\...
  installation complete.
19.
20.
21.                 /* Note on Schemes: The default graphing options in STATA are ugly.
  >                         A user-developed package called schemepack includes many mor
  > e
  >                         visually appealing schemes. The code below installs that pac
  > kage
  >                         and sets the scheme to one I like, but you should check out
  > the
  >                         github page and pick one for yourself
  >
  >                         https://github.com/asjadnaqvi/Stata-schemes */
```

```
22.
23.         ssc install schemepack, replace // Appealing graph schemes (optional)
  checking schemepack consistency and verifying not already installed...
  installing into C:\Users\rejeffri\ado\plus\...
  installation complete.
24.         set scheme white_tableau // Add ", perm" after this command to lock change i
  > n
25. }

26.
27. ********************************************************************************
  > ******
28.
29.
30.
31.
32. **#
33. // ---------------------------------------------------- (1) Loading data and initia
  > l exploration
34. import delimited using "O:\Fall 2021\U of A\MET HW\Final Project\county_census_vax.c
  > sv", clear
  (encoding automatically selected: ISO-8859-1)
  (14 vars, 3,038 obs)

35.
36. // NOTE: Our COVID data is from one day, 11/16/2020.
37.
38. * First thing to do is get a sense of our variables
39. summarize _all, sep(0)
```

Variable	Obs	Mean	Std. dev.	Min	Max
year	**3,038**	**2016**	**0**	**2016**	**2016**
state	**0**				
state_po	**0**				
county_name	**0**				
perc	**0**				
partyy	**0**				
datex	**0**				
death_rate	**3,038**	**.0007086**	**.0011324**	**0**	**.0260361**
case_rate	**3,038**	**.0391432**	**.0456962**	**.000721**	**1.145119**
datey	**0**				
completene~t	**3,038**	**.8142446**	**.2877802**	**0**	**.988**
administer~p	**3,038**	**.3655382**	**.2209842**	**0**	**.999**
series_com~v	**0**				
series_com~t	**3,038**	**.3523756**	**.1734873**	**0**	**.999**

```
40.
41. * Convert string to float
42. destring perc, replace force
  perc: contains nonnumeric characters; replaced as double
  (1 missing value generated)

43.
44. // Often RDDs are motivated by a simple scatterplot with the running variable on the
  >  x-axis and outcome on the y.
45. // Try making such a scatter plot here.
46. twoway scatter case_rate perc, xline(0.5) // Make a simple scatterplot

47.
```

```
48. label variable perc "Percent Republican"

49. gen bin = floor(perc*100) // Creating bins of each percentage
  (1 missing value generated)

50. replace bin = bin + 1 if bin >= 0 // Personal preference not to have a 0 bin
  (3,037 real changes made)

51. bysort bin: egen avg_deaths = mean(death_rate) // Average death rate by bin

52. twoway scatter avg_deaths bin, xline(50) // Make a simple scatterplot

53.
54. ********************************************************************************
  > ******
55.
56.
57.
58.
59.
60. **#
61. // ------------------------------------------------------ (2) Testing for gaming, heap
  > ing and power
62. * We need to check for a few issues that might undermine the continuity assumption.
63. * Firstly, do we see evidence that people "gamed" the cutoff to try to get on one si
  > de?
64. bysort bin: egen count = count(bin)

65. twoway bar count bin, xline(50)

66.
67.
68. * Testing for gaming of the cutoff
69. rddensity perc, c(0.5) plot ///
  > cirl_opt(color(blue%0)) cirr_opt(color(red%0)) /// left and right confidence interva
  > l
  > esll_opt(color(blue%0)) eslr_opt(color(red%0)) /// left and right line
  > histl_opt(color(blue%30) barwidth(.009)) /// left histogram
  > histr_opt(color(red%30) barwidth(.007)) // right histogram
Computing data-driven bandwidth selectors.

Point estimates and standard errors have been adjusted for repeated observations.
(Use option nomasspoints to suppress this adjustment.)

RD Manipulation test using local polynomial density estimation.

     c =     0.500 | Left of c  Right of c          Number of obs =         3037
-------------------+----------------------          Model         = unrestricted
     Number of obs |       586        2451          BW method     =          comb
Eff. Number of obs |       286         425          Kernel        =    triangular
    Order est. (p) |         2           2          VCE method    =     jackknife
    Order bias (q) |         3           3
       BW est. (h) |     0.090       0.090

Running variable: perc.

------------------------------------------------------------------------------
            Method |       T          P>|T|
-------------------+----------------------------------------------------------
            Robust |   -1.9588        0.0501
------------------------------------------------------------------------------
```

P-values of binomial tests. (H0: prob = .5)

Window Length / 2	<c	>=c	P>|T|
0.003	**12**	**8**	**0.5034**
0.006	**24**	**18**	**0.4408**
0.009	**36**	**29**	**0.4570**
0.012	**44**	**36**	**0.4340**
0.015	**55**	**43**	**0.2664**
0.018	**60**	**52**	**0.5085**
0.021	**72**	**64**	**0.5485**
0.024	**85**	**74**	**0.4278**
0.027	**100**	**80**	**0.1565**
0.030	**111**	**88**	**0.1186**

```
70.
71.
72.
73. * Testing for statistical power (benchmark of 0.8)
74. rdpow case_rate perc, c(0.5) // What effect size are we powered to detect?
```

Cutoff c = .5	Left of **c**	Right of **c**
Number of obs	**586**	**2451**
Eff. Number of obs	**255**	**358**
BW loc. poly. (h)	**0.079**	**0.079**
Order loc. poly. (p)	**1**	**1**
Sampling BW	**0.079**	**0.079**
New sample	**255**	**358**

Number of obs =	**3037**
BW type =	**mserd**
Kernel =	**Triangular**
VCE method =	**NN**
Derivative =	**0**
HA: tau =	**0.010**

Outcome: **case_rate**. Running variable: **perc**.

Power against:	H0: tau=	0.2*tau =	0.5*tau =	0.8*tau =	tau =
	0.000	**0.002**	**0.005**	**0.008**	**0.010**
Robust bias-corrected	**0.050**	**0.070**	**0.182**	**0.387**	**0.552**

```
75.
76.
77.
78. ********************************************************************************
  > ******
79.
80.
81.
82.
83.
84. **#
85. // ------------------------------------------------- (3a) Estimation via OLS and i
  > nterpretation
86. gen treat = perc > 0.5 // Define a treatment variable

87. label variable treat "Treatment"

88.
89. * Generate new variables for regressions
```

```
. gen perc_std = perc - 0.5 // normalize the percent around 0
  (1 missing value generated)

. gen treat_perc = treat * perc_std // interaction term
  (1 missing value generated)

. gen perc_std2 = perc_std * perc_std     // for quadratic regression
  (1 missing value generated)

. gen treat_perc2 = treat_perc * perc_std2 // for quadratic regression
  (1 missing value generated)

. gen bins = round(perc_std, 0.01) // Creating bins of each percentage
  (1 missing value generated)

. replace bins = bins + .01 if bins >= 0 // Personal preference not to have a 0 bin
  (2,471 real changes made)

. bysort bins: egen avg_deaths2 = mean(death_rate) // Average death rate by bin

. bysort bins: egen avg_cases = mean(case_rate) // Average death rate by bin

. bysort bins: egen avg_vax = mean(series_complete_18pluspop_pct) // Average death rat
  > e by bin

.
. * Label Variables
. la var death_rate "Death Rate"

. la var case_rate "Case Rate"

. la var series_complete_18pluspop_pct "Vax Rate 18+"

. la var perc_std "% Republican Linear"

. la var treat_perc "Treatment x % Rep."

. la var perc_std2 "% Rep. Quadratic"

. la var treat_perc2 "Treat x % Rep. Sqrd."

.
. * Regressions on death rate
. eststo reg_d1: qui areg death_rate treat perc_std, robust absorb(state) // linear re
  > gression without interaction term

. eststo reg_d2: qui areg death_rate treat perc_std treat_perc, robust absorb(state) /
  > / linear regression with interaction term

. eststo reg_d3: qui areg death_rate treat perc_std perc_std2 treat_perc treat_perc2,
  > robust absorb(state) // quadratic regression

.
. * Regressions on case rate
. eststo reg_c1: qui areg case_rate treat perc_std, robust absorb(state) // linear reg
  > ression without interaction term

. eststo reg_c2: qui areg case_rate treat perc_std treat_perc, robust absorb(state) //
  >  linear regression with interaction term
```

```
117 eststo reg_c3: qui areg case_rate treat perc_std perc_std2 treat_perc treat_perc2, r
  > obust absorb(state) // quadratic regression

118
119 * Regressions on vaccination rate
120 eststo reg_v1: qui areg series_complete_18pluspop_pct treat perc_std, robust absorb(
  > state) // linear regression without interaction term

121 eststo reg_v2: qui areg series_complete_18pluspop_pct treat perc_std treat_perc, rob
  > ust absorb(state) // linear regression with interaction term

122 eststo reg_v3: qui areg series_complete_18pluspop_pct treat perc_std perc_std2 treat
  > _perc treat_perc2, robust absorb(state) // quadratic regression

123
124 * Generate table with both linear and quadratic regressions
125 esttab reg_d*, se label
```

	(1) Death Rate	(2) Death Rate	(3) Death Rate
Treatment	-0.000179**	-0.0000597	0.0000134
	(0.0000623)	(0.0000538)	(0.0000803)
% Republican Linear	-0.000715**	-0.00257**	-0.00690*
	(0.000226)	(0.000945)	(0.00300)
Treatment x % Rep.		0.00240*	0.00942*
		(0.000982)	(0.00416)
% Rep. Quadratic			-0.0136*
			(0.00696)
Treat x % Rep. Sqrd.			0.0195
			(0.0119)
Constant	0.000949***	0.000725***	0.000523***
	(0.0000798)	(0.0000516)	(0.000117)
Observations	3037	3037	3037

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

```
126 esttab reg_c*, se label
```

	(1) Case Rate	(2) Case Rate	(3) Case Rate
Treatment	-0.00490*	-0.000694	0.00376
	(0.00237)	(0.00172)	(0.00303)
% Republican Linear	-0.0310***	-0.0962*	-0.321*
	(0.00888)	(0.0397)	(0.129)
Treatment x % Rep.		0.0845*	0.441*
		(0.0411)	(0.179)
% Rep. Quadratic			-0.707*
			(0.297)
Treat x % Rep. Sqrd.			1.064*
			(0.503)

Constant	**0.0472***	**0.0394***	**0.0289***
	(**0.00334**)	(**0.00158**)	(**0.00477**)
Observations	**3037**	**3037**	**3037**

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

```
127 esttab reg_v*, se label
```

	(1) Vax Rate 18+	(2) Vax Rate 18+	(3) Vax Rate 18+
Treatment	**0.0132**	**0.00644**	**0.00764**
	(**0.00749**)	(**0.00838**)	(**0.00950**)
% Republican Linear	**-0.312*** **	**-0.207*** **	**-0.262**
	(**0.0213**)	(**0.0590**)	(**0.154**)
Treatment x % Rep.		**-0.135***	**-0.0501**
		(**0.0636**)	(**0.239**)
% Rep. Quadratic			**-0.172**
			(**0.484**)
Treat x % Rep. Sqrd.			**0.268**
			(**0.825**)
Constant	**0.384*** **	**0.396*** **	**0.394*** **
	(**0.00494**)	(**0.00743**)	(**0.00931**)
Observations	**3037**	**3037**	**3037**

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

```
128
129
130
131
132
133 **#
134 // ------------------------------------------------- (4) BANDWIDTH TEST on RUNNING
  >  VARIABLE
135
136
137
138 est clear // clear any existing estimations

139
140
141
142 * Death Rate Bandwidth - Linear Fit
143 forvalues h = 0.4(-0.1)0.1 {
    2.          qui areg death_rate treat perc_std treat_perc if abs(perc_std) <= `h', ro
  > bust absorb(state) // linear regression without interaction term
    3.          tempvar i
    4.          gen i = `h'*10
    5.          eststo regd1_`i'
    6.          drop i
    7. }
```

```
144 esttab regd1_*, keep(treat) ti("Death Rates Across Multiple Bandwidths - Linear Fit"
  > ) mtitles("+-40%" "+-30%" "+-20%" "+-10%") se label
```

Death Rates Across Multiple Bandwidths - Linear Fit

	(1) +-40%	(2) +-30%	(3) +-20%	(4) +-10%
Treatment	**-0.0000561**	**0.0000525**	**0.000124**	**-0.0000430**
	(**0.0000550**)	(**0.0000780**)	(**0.000147**)	(**0.0000704**)
Observations	**3020**	**2662**	**1705**	**792**

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

```
145
146 * Death Rate Bandwidth - Quadratic Fit
147 forvalues h = 0.4(-0.1)0.1 {
    2.          qui areg death_rate treat perc_std perc_std2 treat_perc treat_perc2 if ab
  > s(perc_std) <= `h', robust absorb(state) // linear regression without interaction te
  > rm
    3.          tempvar i
    4.          gen i = `h'*10
    5.          eststo regd2_`i'
    6.          drop i
    7. }

148 esttab regd2_*, keep(treat) ti("Death Rates Across Multiple Bandwidths - Quadratic F
  > it") mtitles("+-40%" "+-30%" "+-20%" "+-10%") se label
```

Death Rates Across Multiple Bandwidths - Quadratic Fit

	(1) +-40%	(2) +-30%	(3) +-20%	(4) +-10%
Treatment	**0.0000524**	**0.0000372**	**-0.000149**	**0.0000309**
	(**0.0000846**)	(**0.0000933**)	(**0.0000888**)	(**0.0000956**)
Observations	**3020**	**2662**	**1705**	**792**

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

```
149
150
151
152
153
154
155 * Case Rate Bandwidth - Linear Fit
156 forvalues h = 0.4(-0.1)0.1 {
    2.          qui areg case_rate treat perc_std treat_perc if abs(perc_std) <= `h', rob
  > ust absorb(state) // linear regression without interaction term
    3.          tempvar i
    4.          gen i = `h'*10
    5.          eststo regc1_`i'
    6.          drop i
    7. }
```

```
157 esttab regc1_*, keep(treat) ti("Case Rates Across Multiple Bandwidths - Linear Fit")
  >  mtitles("+-40%" "+-30%" "+-20%" "+-10%") se label
```

Case Rates Across Multiple Bandwidths - Linear Fit

	(1) +-40%	(2) +-30%	(3) +-20%	(4) +-10%
Treatment	**-0.000554**	**0.00464**	**0.00733**	**0.00216**
	(**0.00178**)	(**0.00295**)	(**0.00614**)	(**0.00200**)
Observations	**3020**	**2662**	**1705**	**792**

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

```
158
159 * Case Rate Bandwidth - Quadratic Fit
160 forvalues h = 0.4(-0.1)0.1 {
    2.         qui areg case_rate treat perc_std perc_std2 treat_perc treat_perc2 if abs
  > (perc_std) <= `h', robust absorb(state) // linear regression without interaction ter
  > m
    3.         tempvar i
    4.         gen i = `h'*10
    5.         eststo regc2_`i'
    6.         drop i
    7. }

161 esttab regc2_*, keep(treat) ti("Case Rates Across Multiple Bandwidths - Quadratic Fi
  > t") mtitles("+-40%" "+-30%" "+-20%" "+-10%") se label
```

Case Rates Across Multiple Bandwidths - Quadratic Fit

	(1) +-40%	(2) +-30%	(3) +-20%	(4) +-10%
Treatment	**0.00506**	**0.00443**	**-0.00226**	**0.00217**
	(**0.00324**)	(**0.00363**)	(**0.00319**)	(**0.00254**)
Observations	**3020**	**2662**	**1705**	**792**

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

```
162
163
164
165
166
167 * Vaccination Rate Bandwidth - Linear Fit
168 forvalues h = 0.4(-0.1)0.1 {
    2.         qui areg series_complete_18pluspop_pct treat perc_std treat_perc if abs(p
  > erc_std) <= `h', robust absorb(state) // linear regression without interaction term
    3.         tempvar i
    4.         gen i = `h'*10
    5.         eststo regv1_`i'
    6.         drop i
    7. }

169 esttab regv1_*, keep(treat) ti("Vaccination Rates Across Multiple Bandwidths - Linea
  > r Fit") mtitles("+-40%" "+-30%" "+-20%" "+-10%") se label
```

Vaccination Rates Across Multiple Bandwidths - Linear Fit

	(1) +-40%	(2) +-30%	(3) +-20%	(4) +-10%
Treatment	**0.00610**	**0.00754**	**0.00725**	**-0.0113**
	(**0.00845**)	(**0.00894**)	(**0.0104**)	(**0.0152**)
Observations	**3020**	**2662**	**1705**	**792**

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

```
170
171 * Vaccination Rate Bandwidth - Quadratic Fit
172 forvalues h = 0.4(-0.1)0.1 {
    2.         qui areg series_complete_18pluspop_pct treat perc_std perc_std2 treat_per
  > c treat_perc2 if abs(perc_std) <= `h', robust absorb(state) // linear regression wit
  > hout interaction term
    3.         tempvar i
    4.         gen i = `h'*10
    5.         eststo regv2_`i'
    6.         drop i
    7. }

173 esttab regv2_*, keep(treat) ti("Vaccination Rates Across Multiple Bandwidths - Quadr
  > atic Fit") mtitles("+-40%" "+-30%" "+-20%" "+-10%") se label
```

Vaccination Rates Across Multiple Bandwidths - Quadratic Fit

	(1) +-40%	(2) +-30%	(3) +-20%	(4) +-10%
Treatment	**0.00824**	**0.0111**	**0.00152**	**-0.00295**
	(**0.00960**)	(**0.0106**)	(**0.0130**)	(**0.0196**)
Observations	**3020**	**2662**	**1705**	**792**

Standard errors in parentheses
* p<0.05, ** p<0.01, *** p<0.001

```
174
175
176 ********************************************************************************
  > ******
177
178
179
180
181 **#
182 // ---------------------------------------------------- (5) Sharp RDD Visualizations
183
184 // Figure of Linear Regression -- Death Rate
185         #delimit ;
  delimiter now ;
186         twoway (lfit death_rate perc_std if perc_std<0,clcolor(gs4) lpattern(dash))
  >         (lfit death_rate perc_std if perc_std>=0,clcolor(gs4) lpattern(dash))
  >         (scatter avg_deaths2 bins, msize(medsmall) msymbol(circle) mfcolor(white) ml
  > color(black) mlwidth(thin) xline(0, lcolor(cranberry) lpattern(dash))),
  >         ytitle("Death Rate as of 11/16/2020", size(medsmall) margin(small)) ylabel(,
  > labsize(medsmall)  format(%10.7e)) xtitle("Distance to cutoff", size(medsmall) margi
  > n(small)) xlabel( -.50(.1).50, labsize(medsmall))
  >         graphregion(fcolor(white) lcolor(white)) legend(order(2 "Linear fit" 3 "Perc
  > entage bin") size(medlarge)) plotregion(lcolor(black) lwidth(thin));
```

```
187         #delimit cr
  delimiter now cr
188         graph export "O:\Fall 2021\U of A\MET HW\Final Project\Figures, with Vax\Lin
  > earRegDeath.png", width(1000) replace
  file O:\Fall 2021\U of A\MET HW\Final Project\Figures, with Vax\LinearRegDeath.png
      cannot be modified or erased; likely cause is read-only directory or file
  r(608);

  end of do-file

  r(608);

189 do "\\uem.walton.uark.edu\UEMProfiles_Lab$\rejeffri\RedirectedFolders\Documents\STD2
  > 520_000000.tmp"

190 log close
        name:  <unnamed>
         log:  O:\Fall 2021\U of A\MET HW\Final Project\Adopo_Jeffries_Final_log.smcl
    log type:  smcl
   closed on:  17 Dec 2021, 09:29:14
_______________________________________________________________________________
```